Journal of Power Sources 239 (2013) 680–688

Contents lists available at SciVerse ScienceDirect

Journal of Power Sources

journal homepage: www.elsevier.com/locate/jpowsour

Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods☆

Adnan Nuhic [a,*], Tarik Terzimehic [a], Thomas Soczka-Guth [a], Michael Buchholz [b], Klaus Dietmayer [b]

[a] Deutsche ACCUmotive GmbH & Co. KG, Neue Str. 95, 73230 Kirchheim u. Teck (Nabern), Germany
[b] Institute of Measurement, Control, and Microtechnology, Ulm University, 89081 Ulm, Germany

HIGHLIGHTS

► SOH and RUL estimation of a lithium-ion battery with a support vector machine.
► New method for training data processing by load collectives.
► Estimation accuracy over lifetime approved on real driving profiles.
► Application on-board a battery management system conceivable.

ARTICLE INFO

Article history:
Received 17 September 2012
Received in revised form
26 November 2012
Accepted 27 November 2012
Available online 13 December 2012

Keywords:
Lithium-ion battery
Battery management system
State of health
Remaining useful life
Support vector machine

ABSTRACT

The accurate estimation of state of health (SOH) and a reliable prediction of the remaining useful life (RUL) of Lithium-ion (Li-ion) batteries in hybrid and electrical vehicles are indispensable for safe and lifetime-optimized operation. The SOH is indicated by internal battery parameters like the actual capacity value. Furthermore, this value changes within the battery lifetime, so it has to be monitored on-board the vehicle. In this contribution, a new data-driven approach for embedding diagnosis and prognostics of battery health in alternative power trains is proposed. For the estimation of SOH and RUL, the support vector machine (SVM) as a well-known machine learning method is used. As the estimation of SOH and RUL is highly influenced by environmental and load conditions, the SVM is combined with a new method for training and testing data processing based on load collectives. For this approach, an intensive measurement investigation was carried out on Li-ion power-cells aged to different degrees ensuring a large amount of data.

1. Introduction

The most limiting factor of electric and hybrid vehicles popularization in means of transport is currently the vehicle's battery. The battery increases the price of the vehicle, thus, it becomes more expensive compared to conventional vehicles [1,2]. Because of many advantages, Lithium-ion (Li-ion) batteries are the most used battery type in hybrid and electric vehicles, nowadays [3]. Since this technology is present on the market for only a relatively short period, not all its characteristics are well-known. Gaining more knowledge about battery lifetime behavior would eventually result in the development of cost-effective and long lasting batteries.

However, independent from battery design, environmental impacts and dynamical cycling will always push the battery aging and thereby impede the battery in its maximum performance over lifetime. Therefore, it is always desirable to monitor the underlying degradation to be able to track the actual performance and take countermeasures if developing faults occur. This task is called health diagnosis. A recent summary on methods for Li-ion battery diagnosis can be found in Ref. [4]. Prognostics for batteries, on the other hand, predict the remaining useful life (RUL), i.e. how soon a battery pack component (e.g. cells) will fail or reach a level that cannot guarantee satisfactory performance. Diagnosis and prognostics, therefore, are two integral parts in realizing a battery health monitoring system.

Health monitoring embedding diagnosis and prognostics for machinery has gained much attention in the research community in recent years [5,6]. However, an electro-chemical system is fundamentally different from a mechanical system in various aspects. The electro-chemical reactions inside a Li-ion battery pack

☆ This contribution was originally presented as a poster at UECT 2012.

* Corresponding author. Tel.: +49 (0)15158624085; fax: +49 (0)7113052131657.
E-mail address: adnan.nuhic@daimler.com (A. Nuhic).

0378-7753/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.jpowsour.2012.11.146

are almost inaccessible by using common sensor technologies. Therefore, the most available monitoring data collected from Li-ion batteries are from terminal behavior such as voltage, current, and temperature. Finally, compared to mechanical systems, the operation profiles of Li-ion batteries show much more dynamics. A good example for that is a hybrid vehicle, where the Li-ion battery condition is affected by the driver's behavior and environment. Factors affecting the performance and health of Li-ion batteries include aging-dependent capacity loss, capacity imbalance among battery cells, self-discharge, etc. Therefore, the development of appropriate methodologies and algorithms for monitoring these values have to take into account the uniqueness of Li-ion battery system [7].

The permanent reliable operation of the battery requires these monitoring algorithms to be implemented on-board the vehicle within the battery management system (BMS) [8]. For choosing appropriate algorithms, compromise needs to be made between their complexity and their diagnosis and prognostics accuracy/capability.

Several approaches for on-board suitable algorithms exist today. The usage of model-based tracking methods is a common way to achieve desired results [4]. The usage of Kalman filtering with electro-chemical or electrical equivalent-circuit models for monitoring was reported in a lot of works, e.g. Refs. [9,10]. But multiple sources of errors like sensor offsets, degrading sensor fidelity, or the quality of measured data impede this estimation, especially when it is used on-board a vehicle with reduced computation capabilities. Automated reasoning schemes based on neuro-fuzzy and decision theoretic methods, like Autoregressive Integrated Moving Average (ARIMA), have been investigated for both diagnostics and prognostics tasks [11]. Not disclaiming the work done before, it still remains difficult to accurately monitor the battery health or predict the remaining useful life under arbitrary environmental and load conditions.

At this point, the usage of data-driven methods is convenient due to their ability to transform high-dimensional and noisy environmental data into lower-dimensional information for diagnostics and, especially, for prognostics tasks [12]. In this contribution, a new data-driven approach is developed for embedding diagnosis and prognostics of battery health in automotive applications. For the estimation of SOH and RUL, one of today's most powerful and popular machine learning algorithms, the support vector machine (SVM), is combined with a completely new method for data processing. The input and output vectors of the required SVM learning data set are generated by processing the measured data through load collectives. As the estimation of SOH and RUL is strongly influenced by environmental, ambient, and load conditions, this method processes the data in respect to these dependencies, including even the operation history. Furthermore, to ensure a large amount of training and testing data, an intensive measurement investigation was carried out on automotive Lithium-ion power-cells aged to different degrees.

The following sections will expand more on the chosen algorithm in Section 2, our implementation approach in Section 3, the experimental setup and corresponding results in Section 4, and concludes with a summary in Section 5.

2. Intelligent battery health monitoring

2.1. Defining battery health

Usually, the term state of health (SOH) is used to characterize the battery health status. The SOH describes the physical condition of the battery, which is commonly characterized by the loss of rated capacity:

$$\mathrm{SOH} = \frac{C_{\mathrm{act}} - C_{\mathrm{EOL}}}{C_{\mathrm{nom}} - C_{\mathrm{EOL}}} \cdot 100\% \quad C_{\mathrm{act}} \geq C_{\mathrm{EOL}}. \tag{1}$$

Here, C_{act} is the actual capacity of the battery and C_{nom} represents the nominal capacity of a brand-new battery. For the Eq. (1), an end of life (EOL) capacity C_{EOL} at SOH = 0% has to be defined, too. In the battery manufacturing industry, this value is often reached if the actual capacity drops below 80% of its initial value

$$C_{\mathrm{EOL}} = 0.8 \cdot C_{\mathrm{nom}}. \tag{2}$$

However, the SOH value declines as a function of time through battery usage and aging from 100% to 0%. The number of charge–discharge cycles related to the specific performance (until i.e. 80% of the nominal capacity is reached) is the remaining useful life (RUL) of the battery. In this work, the degradation trend of the time-varying capacity is tracked, and the number of cycles to SOH = 0% is estimated to realize the proposed approach.

2.2. Support vector regression

The support-vector-machines (SVMs) have been applied for classification problems in various domains of pattern recognition. A comprehensive introduction can be found e.g. in Refs. [13,14]. However, the SVM can also be applied to regression problems, although regression is inherently more difficult than classification. The SVM used for regression as a non-linear estimator is more robust than a least-squares estimator because it is insensitive to small changes [15].

Let the training data set be given with $(x_1, y_1),\ldots,(x_l, y_l) \subset X \times R$, where X denotes the space of input patterns (e.g. q-dimensional space $X = R^q$). The goal of ε support vector regression (ε-SVR) is to find a function $f(x)$that has a ε deviation at the most from the target patterns y_i for all training data, while at the same time the function is as flat as possible. In other words, attention is not paid to the errors as long as they are smaller than ε, but also deviations bigger then ε are not accepted. However, sometimes it is not possible to find such a function, or it is desirable to allow some errors. For this purpose, the so-called slack variables ξ_i, ξ_i^* are introduced in order to cope with otherwise unsolvable optimization problem constraints. In the case that the demanded function is linear,

$$f(x) = \langle \omega, x \rangle + b, \omega \in X, b \in R, \tag{3}$$

the optimization problem has the form

$$\begin{aligned} &\min_{\omega} \frac{1}{2}\left\|\omega^2\right\| + C\sum_{i=1}^{l}\left(\xi_i + \xi_i^*\right) \\ &\text{subject to} \begin{cases} y_i - \|\omega, x_i\| - b \leq \varepsilon + \xi_i \\ \|\omega, x_i\| + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0. \end{cases} \end{aligned} \tag{4}$$

The parameter $C > 0$ determines the trade-off between flatness of the function $f(x)$ (i.e. simplicity of the function) and the amount of deviations higher than ε that is tolerated. Tolerating deviations can be represented by the ε-insensitive loss function $|\xi|_\varepsilon$:

$$|\xi|_\varepsilon = \begin{cases} 0 & \text{if} |\xi| \leq \varepsilon \\ |\xi| - \varepsilon & \text{otherwise}. \end{cases} \tag{5}$$

Graphically, this is shown in Fig. 1. Only points outside the shaded areas increase the amount of deviation.

Eq. (4) represents a dual optimization problem which is much easier for solving, and, more importantly, makes it possible to apply

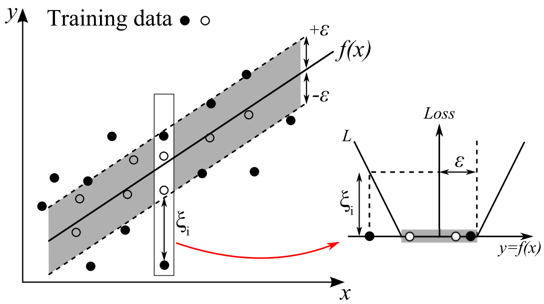

Fig. 1. ε-Insensitive loss function setting for linear SVR [13].

SVRs to non-linear functions. The dual optimization problem is obtained by minimizing the Lagrange function with respect to the primary variables. The first step is the formation of the Lagrange function from the primary optimization problem by introducing a dual set of variables:

$$L := \frac{1}{2}\left\|\omega^2\right\| + C\sum_{i=1}^{l}\left(\xi_i + \xi_i^*\right) - \sum_{i=1}^{l}\left(\eta_i\xi_i + \eta_i^*\xi_i^*\right) - \sum_{i=1}^{l}\alpha_i(\varepsilon + \xi_i - y_i + \langle\omega, x_i\rangle + b) - \sum_{i=1}^{l}\alpha_i^*\left(\varepsilon + \xi_i^* + y_i - \langle\omega, x_i\rangle - b\right). \tag{6}$$

Here, L stands for the Lagrangian and $\eta_i, \eta_i^*, \alpha_i$, and α_i^* are Lagrange multipliers. After minimizing the Lagrange function, the obtained target function $f(x)$ can be written as

$$f(x) = \sum_{i=1}^{l}\left(\alpha_i - \alpha_i^*\right)\langle x_i, x\rangle + b. \tag{7}$$

After training the SVR, the values of α_i and α_i^* are both zero if x_i does not contribute to the loss function. Therefore, only the support vectors of x_i have non-zero values either for α_i or α_i^*:

$$\mathrm{SV} = \left[x_i \middle| \left(\alpha_i > 0 \vee \alpha_i^* > 0\right), i = 1, 2, \ldots l\right]. \tag{8}$$

The big advantage of a properly optimized SVR is its ability to condense thousands of training points to a manageable number of support vectors (SVs). The another benefit after learning the SVs is that the SVR usually does not require matrix inversions and calls to computationally intensive math functions for their operation, which are required by most model-based approaches like Kalman filters.

In the field of SOH estimation, it is possible to design an SVR that is able to incorporate a large amount of training data points and reduce them to a set of SVs that can be manipulated also with low computational capabilities. The key to use the full potential of SVs is to choose the right training data and proper kernel functions. In our work, we use libSVM [16] to determine the SVs with variations of its input training vectors, so the resulting SVR can automatically be tested for accuracy.

3. Battery health estimation approach

In this section, the steps applied for training an SVR are presented, followed by our SOH & RUL estimation approach. The training steps include: data preprocessing, composing training data, and search for the optimal SVR parameters. The latter step is explained together with the total estimation approach.

3.1. Data preprocessing

Data preprocessing turns out to be the key in getting the SVR to converge. Without preprocessing, a lot of training attempts do not result in a converging SVR. The first step of preprocessing consists of scaling the data such that all input vector elements are in the range of -1.0 to 1.0. Without scaling, it could happen that some vector elements, also called features, dominate over the others and push the SVR to converge to an unsatisfactory result [17]. A scaled and representative input vector for training is shown in Table 1.

The second data preprocessing step reduces the dimensionality of the input vector. As large matrices are used for input, they may contain irrelevant or redundant information that may elongate the training process immensely. To prevent this, various methods for feature reduction can be used like the principal component analysis (PCA) or genetic algorithms. However, these rather complex methods have to be used with care due to the fact that they also can exclude important features which are required for a successful convergence of the SVR. Therefore, a more simple method for feature reduction proposed in Ref. [18] was used in this work. The Fisher ratio

$$\mathrm{FR}(x_i) = \frac{\left[\mathrm{mean}\left(x_i^+\right) - \mathrm{mean}\left(x_i^-\right)\right]^2}{\mathrm{var}\left(x_i^+\right) - \mathrm{var}\left(x_i^-\right)} \tag{9}$$

is a method primary used in classification problems but also can be transferred to regression [18]. For continuous values in the input vectors it is suggested to separate the values of the vector into upper 50% (x_i^+) and lower 50% (x_i^-) of values. By eliminating small ratio values (FR < 2), Eq. (9) gives an information on whether a particular attribute x_i affects the regression result or not.

3.2. Training data composition

For composing good training data, several criteria should meet as follows:

- At first, the training data should be different from the data that will be used later for testing and validation. If training data and testing data are identical, the SVR tends to interpolate just points on a line during testing which is not what we expect the SVR to do. Also, the resulting SVR can be "overfitted" and therefore not performing well for real world data different from the training data.
- Training data should cover the expected range of operation of the final SVR implemented on-board the vehicle.
- Training data should be compatible to the vehicle's central controller unit dependent data structure for registering measured battery values.

Table 1
An example of an input data vector.

Element	SOC_0	ΔC	Ah	Cycle number	Temperature	Time
Unscaled	70	0.07439	6637.42	564	27.7326	2410036.65
Scaled	0.6923	-0.0299	0.9834	-0.8737	-0.0072	0.5221

Based on the last item, this work introduces a completely new method for composing the training data by processing them with load collectives. This is a convenient way, because many of today's vehicle central controller units collect load cycles of the engine, gear drive, drive axles, or also the high voltage (HV) battery in hybrid or electrical vehicles. This type of battery signal monitoring is not demanding in terms of memory, unlike the continuous archiving of appropriate signal changes (i.e. data logging). Load collectives provide information about the occurrence frequency of a certain combination of two signals (e.g., current or state of charge and temperature) during battery cycling. This type of load cycle counting of two signals is also called two parameter instantaneous-value (dwell time-) counting [19]. For Signal 1 from Figs. 2 and 7 classes are defined, as well as for the second signal. To each field of a combination of two signals, a counter is assigned that increments when the value of signal samples are in the appropriate range of classes that belong to this field. Looking at the last sample in Fig. 2, it is seen that the value of the signal 1 is within class 7 and the value of the signal 2 is within class 3. Therefore, the counter for combination 7–3 will be increased.

Another type of load cycle counting was also used in this work, the so-called rainflow counting [20]. This algorithm is used in this work for state of charge (SOC) cycle counting, to count e.g. how often the SOC jumps from one value to another and vice versa. A good way to describe is to imagine rain drops falling down a pagoda style roof. If the time axis in Fig. 3 is rotated by 90° in a clockwise direction, then the dotted lines on the left side of the curve arise at its maximum (horizontal view of curve), flow to its next minimum (e.g. $\overline{AB}$) and drop down. Rainflows from a greater maximum source interrupt rainflows from a smaller maximums (e.g. $\overline{CB'}$). Also a rainflow is stopped when it meets a minimum or maximum (horizontal view) that is beneath the starting rainflow source (e.g. $\overline{BC}$, because the minimum at D is beneath the starting minimum). A half-cycle is counted between a maximum and minimum of one line, e.g. $\overline{EF}$ in Fig. 3. This has to be done analogously also for the right of the curve, only that now sources are the minimums (horizontal view of curve). The resulting cycle is a combination of two half cycles from the left and right side of the curve, with the same maximum and minimum values. In Fig. 3, these are A–D–G, E–F–E' and B–C–B', and the counter will be increased only for the correspondent classes 1–7, 2–5 and 4–6.

Now, with the presented cycle counting methods, the training vector composition with load collectives can be introduced, see Fig. 4. The exemplary shown SOC profile was employed to generate the already mentioned SOC rainflow load collective. In addition to this, further load collectives of type temperature over current and SOC over temperature were generated. The arranged matrices were transformed to stacked $[q \times 1]$ vectors and queued to a large input training vector. Beside the cycling periods during the experimental investigation, which is explained in more detail in the next section, capacity tests C_1, C_2 and C_3 (SOC cycles 100–0%) were repetitively performed for performance monitoring. The estimated values of capacity where then stored in the $[q \times 1]$ training output vectors:

$$x_i = \underbrace{\begin{bmatrix} C_1; LC_1 \\ C_1; LC_1 + LC_2 \\ C_2; LC_2 \\ \vdots \end{bmatrix}}_{\text{input training vector}(q\times 1)} \rightarrow \underbrace{\text{Eq.}(6)}_{\text{SV}\rightarrow\text{Eq.}(8)} \leftarrow y_i = \underbrace{\begin{bmatrix} C_2 \\ C_3 \\ C_3 \\ \vdots \end{bmatrix}}_{\text{target training vector}(q\times 1)} . \tag{10}$$

To increase the number of training vectors, the combination of load collectives (LC) and the coherent capacity tests shown in Fig. 4 was varied. For example, cycling load in form of LC_1 between capacity tests C_1 and C_2, where C_1 is the initial capacity is stored together with LC_1 in the first input vector while capacity C_2 is the target value stored in the first output vector. The LC_2 between tests C_2 and C_3 is then added to LC_1 and stored together with C_1 in the second input vector, while C_3 is the second target, and so on. Eq. (10) also clarifies the role of the SVR in SOH & RUL estimation: During the training process, the SVR tries to establish a relationship between the load the battery has experienced and corresponding capacity fade. Furthermore, with the described method for extending the number of training vectors, this relationship is being strengthened. The target of the training process is to extract only the required support vectors (SV) from the SVR in the end.

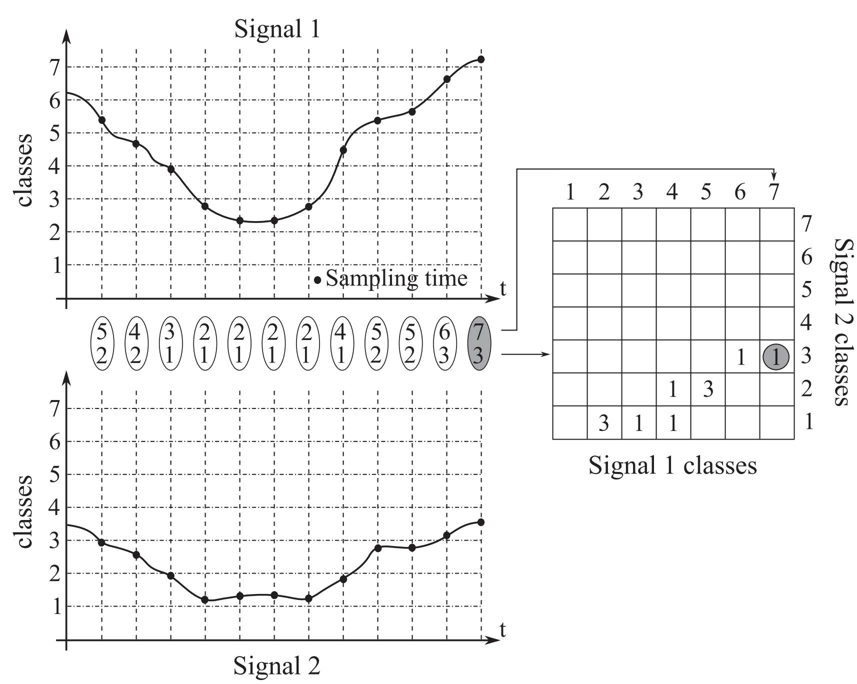

Fig. 2. Two parameter (2D) instantaneous-value (dwell time-) counting.

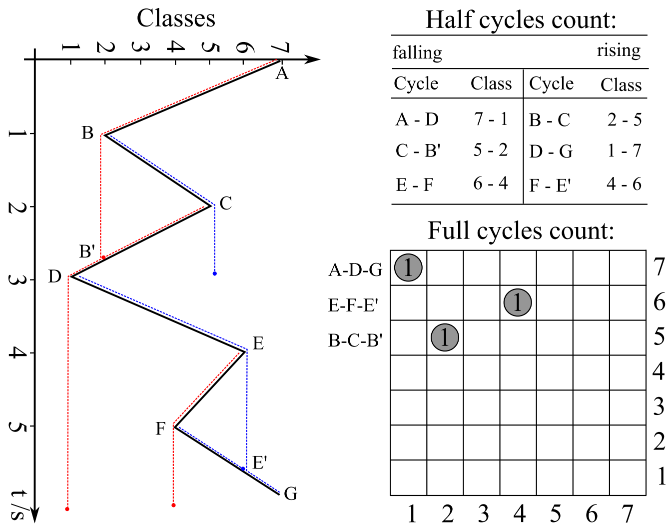

Fig. 3. Rainflow counting.

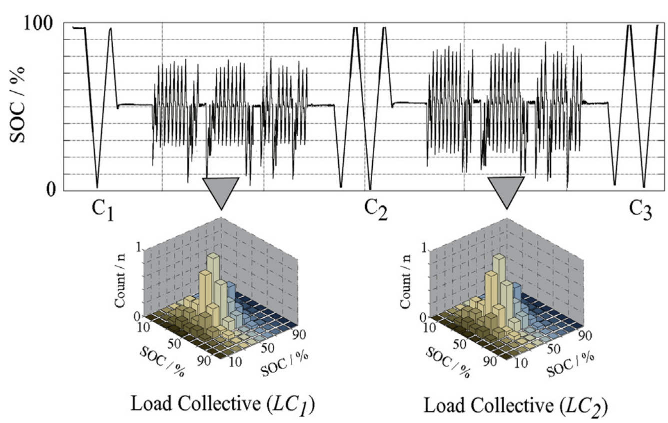

Fig. 4. Generating load collectives from driving profiles (i.e. Rainflow counting for a SOC profile).

To acknowledge the considerable advantages by combining load collectives with SVR for SOH & RUL estimation, a conventional training data set was composed based on battery signals of time transient type. The exemplary SOC profile from Fig. 4 can be processed for the first training data set by calculating the throughput of the cell (Ah) by integrating the current between the first and second capacity tests by

$$\mathrm{Ah} = \int_{C_1}^{C_1} i(t)\mathrm{d}t. \tag{11}$$

This sort of input training vectors x_i also contain information about the capacity value at the beginning of the profile C_1, the min. and max. value for SOC, and temperatures during cycling. Furthermore, they can contain the last known capacity degradation and the time between the initial and target capacity measurements. The first training target vector y_i contains only the cell capacity at the end of driving profile (e.g. C_2). Eq. (12) shows this conventional training data set. The second training data set is then starting after the second capacity test and the further data sets can also variate as described in the previous paragraph for load collectives.

$$x_i = \underbrace{\begin{bmatrix} C_1 \\ \mathrm{Ah} \\ \mathrm{SOC}_1 \\ \mathrm{SOC}_2 \\ T_1 \\ T_2 \\ \vdots \end{bmatrix}}_{\text{input training vector}(p\times 1)} \rightarrow \underbrace{\text{Eq. (6)}}_{\text{SV}\rightarrow\text{Eq. (8)}} \leftarrow y_i = \underbrace{\begin{bmatrix} C_2 \\ \vdots \end{bmatrix}}_{\text{target training vector}(p\times 1)} \tag{12}$$

3.3. Optimal SVR parameter determination

The SVR parameters are the constant C, the size of the error tube ε and the type selection of the kernel function. For that, in case of lower dimensionality p of input vector, the best solution is obtained by using a non-linear support vector regression, especially with the Gaussian kernel [13]. However, if the input space is higher dimensional, as is the case when using load collective data ($q \gg p$), then there is no need to map the original input space into a more dimensional space to find the appropriate regression function. Even more, using complex functions such as Gaussian support vector regression with multidimensional data will result in over-fitting. Therefore, the linear SVR kernel is found to be the best choice for our work. Regarding a later implementation on-board a vehicle with reduced computational possibilities, this choice is additionally advantageous. The parameters C and ε are selected purely empirical, using the cross-validation method of the results according to Ref. [21].

3.4. Estimation of SOH & RUL

The idea behind our SOH & RUL estimation approach is to develop an estimation method that, in its functionality, can immediately be integrated on-board the vehicle. As a real word example, a hybrid or electrical vehicle stresses the HV-battery for a period of one year. After this, a SOH estimation has to be performed e.g. periodically every n-th year. For that, the gathered load collective is loaded in the input vector together with the nominal value of the battery capacity which is estimated during production and stored on the central control unit of the vehicle. This unit also contains the support vectors which have been estimated in an

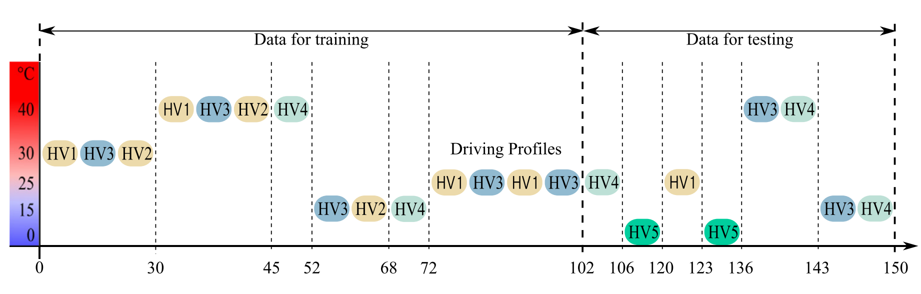

Fig. 5. Cell testing timetable.

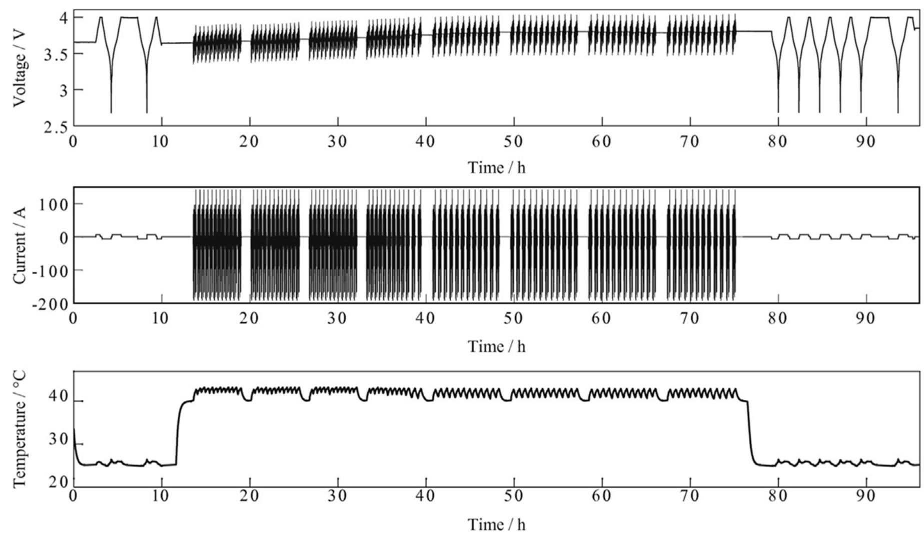

Fig. 6. One of the applied test procedures.

off-line process as described in the previous chapter. By means of the provided load collective, the implemented SVR through Eq. (7) is able to estimate the actually capacity which results after stressing the battery by operation during the last year ($n = 1$). This process is then repeated e.g. next year again ($n = 2$), whereat the load collectives of the first year LC_1 is added to the load collective of the actual, second year LC_2, and so on. Finally, the predicted actual capacity is transferred to an SOH value in % by Eq. (1):

$$
\begin{aligned}
&n = 1, 2, \ldots m \text{ years}\\
&x = \underbrace{\left[C_{\text{norm}};\ \sum_{n=1}^{m} LC_n\right]}_{\text{input vector}(q\times 1)} \rightarrow \underbrace{\left[x_i = x_1^{\text{SV}}, x_2^{\text{SV}}, x_3^{\text{SV}}; \ldots\right]}_{\text{support vectors}\ (q\times 1)\text{from Eq. (8)}}\\
&\rightarrow \underbrace{f(x_n)}_{\text{Eq. (7)}} = \underbrace{\left[C_{\text{act},n}\right]}_{\text{output vector}(1\times 1)}
\end{aligned} \tag{13}
$$

Using the same estimation method, it is possible to achieve a simple battery state prognostic, which means to estimate the value of the battery capacity in the near or distant future. As a future battery load is not known, one option is to take as a reference the last n loads (last n load collectives) archived in the memory of the central control unit. From these matrices, the mean load value could be calculated in order to obtain a reference load per time unit. The product of reference load with a time interval on which we want to predict the state of the cell is used as the input vector of the SVR and thus allows an estimation of the battery state in the future.

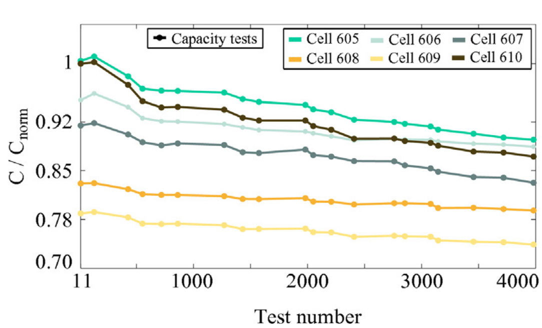

Fig. 7. Capacity fading during testing.

$$
\begin{aligned}
&n = 1, 2, \ldots m \text{ years}\\
&x = \underbrace{\left[C_{\text{norm}};\ \sum_{n=1}^{m} LC_n\right]}_{\text{input vector}(q\times 1)} \rightarrow \underbrace{\left[x_i = x_1^{\text{SV}}, x_2^{\text{SV}}, x_3^{\text{SV}}; \ldots\right]}_{\text{support vectors}(q\times 1)\text{from Eq. (8)}}\\
&\rightarrow \underbrace{f(x_n)}_{\text{Eq. (7)}} = \underbrace{\left[C_{\text{act},n}\right]}_{\text{output vector}(1\times 1)}\\
&\rightarrow C_{\text{act},n} \le 0.8 \cdot C_{\text{nom}} \rightarrow n \text{ years}
\end{aligned} \tag{14}
$$

Another option is not to use the last n loads, but selecting the most intense n loads from which the reference load could be calculated in the same way as already explained. In this case, the worst case scenario of cell degradation is predicted. Further option would be using the gathered load collective over the last year n-times to prognosticate the future battery capacity n-years ahead and therefore the SOH. For that, the gathered load collective from the last year is multiplied n-times for $n = 1,2,\ldots$ years and provided to the SVR. For every n the SVR prognosticates the capacity and this process is repeated until estimated value is equal or goes under 80% of the nominal capacity. This number of repetitions n represents then the remaining useful life (RUL) in years based on the past battery load.

4. Results

4.1. Experimental investigation

A lot of presented SOH & RUL estimation approaches in literature lack of applicable measured data for validation. Most of the accomplished investigations are based on very uniform tests, batteries are cycled only at one or a few values of SOC, temperature, and/or depth of discharge (DOD), so that the obtained approaches are at least not validated for an application in real world dynamical situations if not useless. To develop practical models, tests must be, as much as possible, similar to real driving profiles, which means that they have to contain different operating conditions (temperatures, SOC, DOD and C-rates).

For this work, six high power Li-ion cells for automotive application in hybrid vehicles were used for the experimental investigation. The cells were taken from the same batch after production, whereat three cells already experienced cycle aging before our

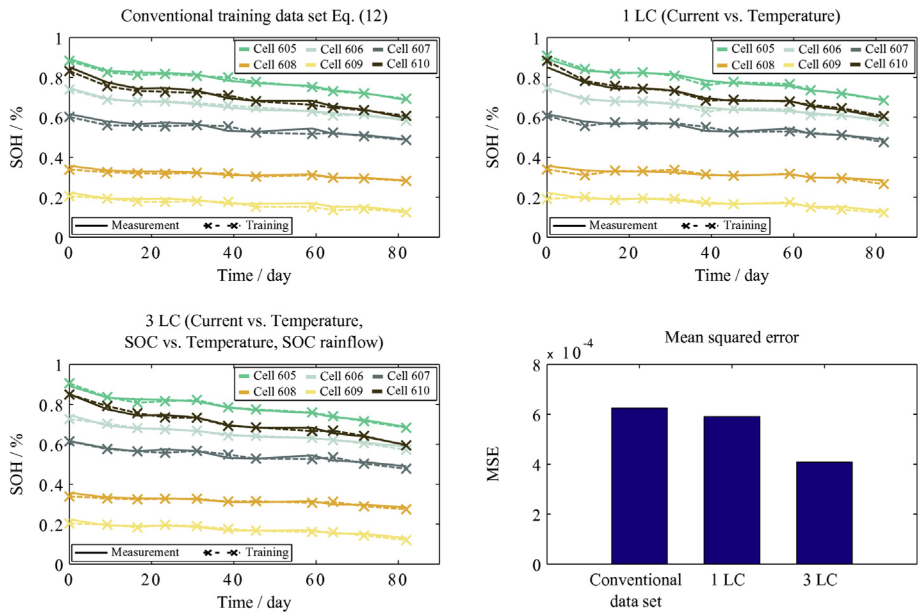

Fig. 8. Results in SVR training by using the conventional data set, only one LC and all provided LCs.

investigation and the other three cells were stored at room temperature and therefore show higher capacities. During the test procedure, the cells operated in temperature chambers so that different temperature operation conditions could be realized. For a period of six months and starting at different aging conditions, the cells were then cyclically stressed with real world driving profiles (HV1 – HV5) which were recorded in different Mercedes-Benz hybrid vehicles during road endurance tests. An overview of the testing procedure is shown in Fig. 5.

An automated battery tester was used and programmed with different test scenarios. To obtain operating conditions as close as possible to real operation in vehicles, the test procedures contain different temperatures, C-rates, SOCs and/or DODs. Every test scenario starts with a full cycle, which implies one full discharge and one full charge, with 1 C current. After a relaxation time, cells are discharged to a certain starting value of SOC. During 6 months of testing, test procedures started with different SOCs from 40% to 80%. Also, operating temperature was changed during testing in the range from 0 °C to 40 °C as shown in Fig. 5. After initial SOC and operating temperature were set up, one of the five driving profiles HV1 – HV5 was applied to the cells with a different number of cycles and different resting times in-between. Upon completion of cycling, temperature was set back to room temperature (25 °C), and after a relaxation time, testing was continued with the next test scenario. Every few cycles, capacity tests were made (vertical dashed lines in Fig. 5), so the cells performances could be monitored during the whole testing time. Each charging and discharging is done with the constant current constant voltage (CCCV) method. One of the applied test procedures with capacity tests in the beginning and the end of cycling is shown in Fig. 6.

The resulting capacity trend after six months of intensive cell cycling tests is shown in Fig. 7. The results show that some of the driving profiles were more intensive and affected capacity fade more, while others were quite mild.

4.2. Training results

As the capacity degradation is a slow process, and also to reduce the impact of artificial tests signals to the cell performance, only 18 capacity tests were conducted (dots in Fig. 7). According to the recommended procedure methods of machine learning to divide the available data into 2/3 data for training a 1/3 data for testing, eleven data sets (combination of cycling load and ensuing capacity test) were separated for training and seven data sets for testing. After employing the method described in Section 3.2 for increasing the number of training data sets, a total of 66 samples per cell or 396 samples for all cells is included in the data set. In the case when the training data is generated with the data set of Eq. (12), the input vector is of dimension $p = 8$, i.e. the input vector of this data set consists of 8 attributes (values). When training data is composed with load collectives, i.e. the load collective matrix "current over temperature" with dimension 7×13, "SOC over temperature" with dimension 7×9, and the SOC rainflow matrix, the training input vector has 274 attributes, of which 214 are non-zero during training. The results of the training process with the conventional data set, with only one load collective, and with all load collectives incorporated are shown in Fig. 8. It can be seen that the best training results are achieved with the usage of all available load collectives. This could be explained through the large number of attributes containing relevant information for the SVR when using load collectives compared to the training data set from Eq. (12) with just eight attributes.

4.3. Testing results of SOH & RUL estimation

Since the first eleven driving cycles were used for training, the remaining seven are utilized for testing and therefore for validation of the SVR. During these last seven cycles, the cells were cycled with driving profiles that were not present in cycles during the training phase (Fig. 5). Additionally, cycling the cells at 0 °C occurs only in the testing data. In this way, it is examined how well the developed model generalizes, i.e. how it performs on previously unseen data. Again, the number of testing data has been increased from 7 to 28. Out of 396 training data values, the SVR uses 77 as support vectors. For testing, four cells were selected, whereat one cell is at the begin and one is at the end of its life, while two cells are somewhere in between. Fig. 9 shows a performance comparison of the estimation when the training is done on the conventional data set and on one, or all three load collectives as input. Again, the best testing results are achieved by the usage of all three load collectives. Although the

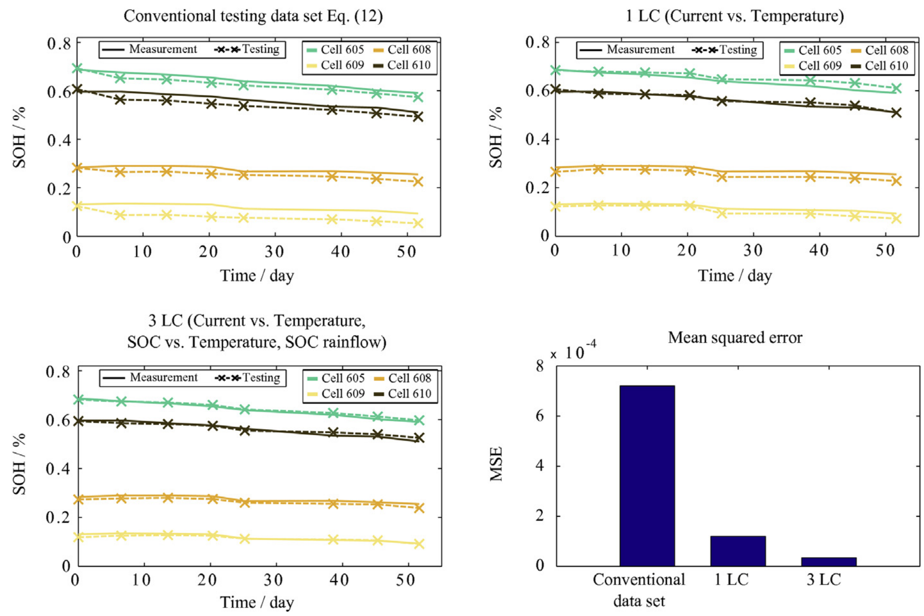

Fig. 9. Results in capacity (SOH) estimation with the SVR by using the conventional data set, only one LC and all provided LCs.

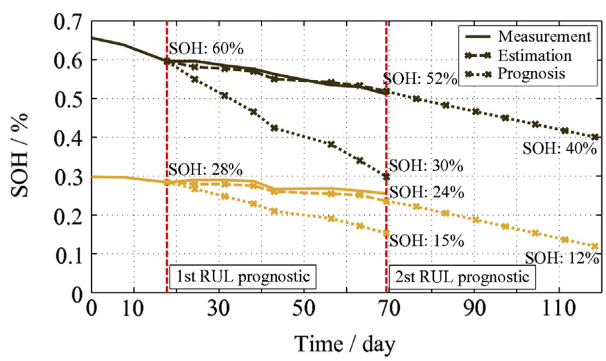

Fig. 10. RUL estimation for a new (brown curve) and already aged cell (orange curve) for the next 50 days (based on load before 1st prognostic). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

testing part contains the profile HV5, which was not part of the training process, the performance of the SVR in estimating the actual capacity remains unaffectedly good.

In Fig. 10, the results of a simple RUL prognostic with the SVR are presented. After the training is conducted on the first eleven training points, at the beginning of the 12th cell testing interval the RUL prediction is started. Thereby, the last cell load is taken as a reference, i.e. the load of the 11th testing interval (before the 12th capacity test). As a very intense profile was driven during the this interval, RUL prognostic for the next 50 days is quite pessimistic compared to the actual state of the cells at the time of the second RUL prognostics. During this in-between period, the cells were cycled with much milder profiles. The second RUL prognostic then uses the load of this last 50 days as a reference. The cell cycling load in this period is not so intense as the one before, so more optimistic RUL prognostic is made, i.e. curve of the degradation prognostic is less steep.

5. Conclusions

A new data-driven approach for embedding diagnosis and prognostics of battery health for automotive applications was developed and validated by real driving cycles in this contribution. For that, one of today's most powerful and popular machine learning algorithms, support vector regression (SVR), was used. By appropriate training it was possible to learn the degradation behavior of Li-ion cells. The validation showed very satisfactory results in the diagnosis of the cells state, especially if one takes into account that the method is validated on driving profiles and temperatures that were not present during training. It is also shown that the developed estimation method can be used for simple RUL prognosis. For more powerful prediction, it would be necessary to include the prediction likelihood in the result, which is not possible with SVRs. For this purpose, as a basis for further work, a relevance vector machine (RVM) may be used, which bases on the same idea as SVR, but outputs a probability density function in addition.

References

[1] L. Lam, A Practical Circuit-Based Model for State of Health Estimation of Li-ion Battery Cells in Electric Vehicles, Master's thesis, University of Technology Delft, Faculty of Electrical Engineering, Mathematics and Computer Science, 2011.

[2] J. Groot, State-of-health Estimation of Li-ion Batteries: Cycle Life Test Methods, Master's thesis, Chalmers University of Technology, Göteborg, Sweden, 2012.

[3] A Study on the Costs and Benefits of Hybrid Electric and Battery Electric Vehicles in Ireland: Hybrid Electric and Battery Electric Vehicles, Technology, Cost and Benefits, Tech. rep., Sustainable Energy Ireland, 2007.

[4] J. Zhang, J. Lee, Journal of Power Sources 196 (2011) 6007–6014.

[5] A. Jardine, D. Lin, D. Banjevic, Mechanical Systems and Signal Processing 20 (2006) 1483–1510.

[6] J. Sikorska, M. Hodkiewicz, L. Ma, Mechanical Systems and Signal Processing 25 (2011) 1803–1836.

[7] S. Santhanagopalan, J. Stockel, R. White, Encyclopedia of Electrochemical Power Sources, Elsevier B.V., 2009 (Ch. Lifetime Prediction).

[8] V. Pop, H.J. Bergveld, D. Danilov, P.P.L. Regtien, P.H.L. Notte, Philips Research: Battery Management Systems - Accurate State-of-charge Indication for Battery-powered Applications, Springer, 2010.

[9] G.L. Plett, Journal of Power Sources 134 (2004) 252–292.

[10] G.L. Plett, Journal of Power Sources 161 (2006) 1356–1384.

[11] J. Kozlowski, Electrochemical cell prognostics using online impedance measurements and model-based data fusion techniques, Aerospace conference, 2003, in: Proceedings of 2003 IEEE, vol. 7, 2003, pp. 3257–3270.

[12] A. Widodo, M.-C. Shim, W. Caesarendra, B.-S. Yang, Expert Systems with Applications 38 (2011) 11763–11769.

[13] A.-J. Smola, B. Schölkopf, Statistics and Computing 14 (3) (2004) 199–222.

[14] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery.
[15] T. Hansen, C.-J. Wang, Journal of Power Sources 141 (2005) 351–358.
[16] C.-C. Chang, C.-J. Lin, Libsvm: A Library for Support Vector Machines. http://www.csie.ntu.edu.tw/cjlin/libsvm (accessed 02.02.12).
[17] C. Edwards, B. Raskutti, The effect of attribute scaling on the performance of support vector machines, Tech. rep., Telstra Research Laboratories, Victoria, Australia.
[18] J. Aldrich, Statistical Science 20 (2005) 401–417.
[19] McKeighan, N. Ranganathan, Fatigue Testing and Analysis Under Variable Amplitude Loading Conditions, ASTM International, 2005.
[20] J. Dambrowski, S. Pichlmaier, A. Jossen, Mathematical methods for classification of state-of-charge time series for cycle lifetime prediction, in: Advanced Automotive Battery Conference Europe, 2012.
[21] A.W. Moore (Ed.), Cross-validation for Detecting and Preventing Overfitting, Carnegie Mellon University, 2001.